PlenumHub / Polyxan

Toward a Source-First Publishing Substrate

Flyxion
Independent Researcher

July 2026

Abstract

This essay outlines a proposed publishing architecture within the PlenumHub/Polyxan/Spherepop lineage, one that treats structured source material rather than any rendered artifact as canonical. Video, audio, prose at varying depths, comics, and interactive renderings are all generated views of an underlying, addressable, branchable state history. The essay argues that most of the epistemic and practical failures of contemporary video-based publishing — oversimplification, unaccountable pronouncement, uncorrectable error, loss of provenance — are consequences of a specific and correctable design decision: conflating source and rendering into a single frozen artifact. It situates this claim within the broader stack, in which Spherepop supplies event histories and provenance, Polyxan supplies addressable semantic and hypermedia structure, and PlenumHub supplies coordination and governance — and argues that source-first publishing is not a new addition to that stack but a straightforward instance of its existing commitments. It closes by proposing a public, competitive practice — a form of *pipeline golf* — in which participants compete to produce the most interesting, entertaining, or explanatorily faithful transformations of a fixed text into image, animation, or sound.

1 The Conflation Problem

Traditional publishing systems, digital and physical alike, generally conflate two things that need not be conflated: the *source* of a work and its *rendering*. A printed book had to be both, because there was no practical channel for distributing structure separately from its typeset form. Contemporary video platforms inherited this fusion by default rather than by necessity. A YouTube video is simultaneously the argument and the recording of the argument; there is no accessible, addressable, versioned object underneath it that a viewer, an editor, or the author's future self can revise, diff, branch, or query.

This conflation produces several downstream pathologies. A mistake in a video is *uncorrectable*: it is permanent unless the entire artifact is withdrawn and re-recorded, and even then, prior copies persist independently in caches and reposts. Because playback is time-locked rather than reader-paced, the medium imposes a kind of *forced linearity*: caveats, branches, and counterexamples — the very content that often carries the most information in theoretical work — are systematically pruned in favor of singular, closure-seeking narratives. There is a *loss of provenance*, since two versions of a talk are unrelated artifacts with no diff between them; there is no equivalent of tracked changes or an errata sheet. And because the medium resists rebuttal-in-kind, an *asymmetry of authority* sets in: claims — particularly interpretive claims about an audience's own psychology — become effectively unfalsifiable within the format that delivered them.

None of these follow from the use of speech or moving images as such. They follow from treating the rendering as the master copy.

2 The Core Inversion

The proposed correction is structural, not stylistic: *the canonical layer is the source; every video, audio track, comic panel, and prose summary is a compiled view of it.* This is the relationship that already holds, uncontroversially, between source code and a compiled binary, between a musical score and a performance, or between a statute's text and its citations. Publishing platforms built around audiovisual media are nearly alone in inverting this relationship.

2.1 Canonical layer

The canonical layer consists of a state or event graph of structured, addressable, diffable units of content, together with the definitions, references, and metadata attached to those units, and a full revision history that is retained rather than overwritten.

2.2 Derived layer

The derived layer consists of everything generated from that canonical layer: prose at variable depth, whether full, medium, or short; audio narration; video, at variable pacing and visual style; comics, slide decks, and interactive simulations; and machine-generated transcripts and descriptions of any of the above, closing the loop back to text.

Correction, branching, and merging all happen exclusively in the canonical layer. Every derived artifact is disposable and regenerable; none of them needs to be hand-edited, because none of them is authoritative.

3 Three Independent Axes

A recurring error in early sketches of this system is collapsing several genuinely independent dimensions into a single notion of "version." At minimum, three axes must be tracked separately:

At minimum, three axes must be tracked separately. The first is *lineage*: revisions, meaning-preserving corrections or extensions written $N \to N + 1$, as distinct from branches, divergent and mutually non-superseding continuations from a common ancestor. Lineage is therefore not a line but a graph. The second is *depth*: deliberately lossy compression of a fixed lineage point — full, medium, short — independent of how recent or how correct that point is. A shorter rendering is not a newer one; a newer revision is not necessarily a shorter one. The third is *rendering*: the output modality, whether video, audio, comic, prose, or transcript, applied to a fixed lineage-and-depth coordinate.

A well-formed request to the system is therefore a coordinate across all three axes: *"the short video of the branch in which the star becomes a planet, as of revision 8."* Treating these as one axis — as conventional platforms implicitly do, where "newer" and "shorter" and "different medium" are all just "another upload" — is precisely what makes existing systems hard to search, cite, or reason about.

4 Toward an Event Grammar

The state graph is only as useful as the precision of its units. Natural-language description — "the yellow triangle transforms into a green star" — is legible to humans but leaves duration, interpolation, persistence, and causation unspecified, and is therefore poorly suited to serve as the actual canonical layer. A structured event, by contrast, is machine-precise and diffable:

```
event {
    target: object_1
    time: 12.0
    shape: star
    color: green
    transition: morph
    duration: 2.0
}
```

Prose, in this architecture, is *generated from* the event, not the reverse: "At 12 seconds, the yellow triangle morphs into a green star over two seconds." The animated-shape case is a useful minimal illustration because its event grammar is nearly trivial to specify by hand. What such toy examples leave open, at the level of this essay, is what an analogous event grammar looks like for discursive, definition-driven theoretical prose, where the "events" are not spatial transformations but the introduction, refinement, or contrast of distinctions.

This is not, in fact, an unsolved problem for the wider program of which this essay is a part. Section 8 develops the necessary machinery directly: a typed calculus of semantic transformations, a closure operator guaranteeing that renderings are always recoverable from a source, an entropy-bounded merge operator for reconciling branches, and a coherence condition guaranteeing that locally acceptable renderings do not silently contradict one another when assembled globally.

5 Provenance Without Rigidity

A system that makes correction too easy risks destroying the very citability it was meant to preserve. If a document's identity is untethered from any fixed point — if paraphrase silently re-anchors what a given reference resolves to — then nothing can be quoted, cited, or held stable against later revision. The intended design resolves this the way version-control systems resolve it: a human-facing name ("the essay," "Document A") always resolves to a current state, but every prior state through which the document has passed remains individually visible and addressable in an unbroken list. Nothing is overwritten. "Show me the full text as it existed before the last revision" must always be answerable.

6 From Publishing System to Media Operating System

Once audience participation is admitted — suggestions, alternate continuations, competing branches — the model stops being a document with history and becomes a *history space*: a graph rather than a line, traversed rather than read. A video renderer follows one path through the graph; a comic renderer may follow another; an exploratory renderer may expose the branching structure itself rather than collapsing it. This is the point at which the system converges, structurally, with source control, game engines, screenplay systems, and knowledge graphs — not by analogy, but because all of these are prior solutions to the same underlying problem: separating representation from rendering.

7 Pipeline Golf

A subsidiary and more immediately playable component of this vision is a public competitive practice, in the spirit of vim golf or code golf, applied to text-to-image and text-to-video pipelines. A fixed source text — a short passage, a single paragraph, an event sequence — is given to participants, who each construct a distinct rendering pipeline for it.

Submissions are judged not on a single axis but several, potentially including fidelity — how much of the source's structure survives the transformation; economy — how few steps, tokens, or how little compute the pipeline requires; explainability — how legible the pipeline's own logic is to an outside reader, whether someone else can read the pipeline and understand why it produced what it produced; and entertainment or surprise — whether the resulting rendering is interesting independent of its fidelity, in the way an unusual translation or an eccentric adaptation can be valuable precisely for departing from its source.

This practice serves two purposes beyond its own entertainment value. First, it is a standing, low-stakes proving ground for the rendering layer of the larger system, generating a continuously growing library of pipelines against which the canonical-source architecture can be tested. Second, it directly operationalizes the essay's central claim: that a single source can and should support many legitimate, non-competing renderings, and that the interesting design space lies in the diversity and quality of the transformation, not in the enforcement of one official version.

8 A Formal Substrate for Canonical Sources

This section develops, from first principles and without appeal to outside material, the minimal formal machinery needed to make the claims of Sections 1–5 precise: that a canonical source can be defined independently of any rendering, that every rendering is recoverable from it, that branches can be reconciled rather than merely coexist, and that the collection of all admissible renderings of a source cannot silently contradict itself as it scales.

8.1 Semantic Objects

Definition 8.1 (Semantic Object)**.** A semantic object is a tuple $\sigma = (I, T, M, E, S)$ where I is an immutable identity, T is a finite set of required modality tags (e.g. PROSE-FULL, PROSE-SHORT, AUDIO, VIDEO, COMIC), $M : T \rightharpoonup$ Content is a partial content map assigning payloads to modalities, $E \in \mathbb{R}_{\geq 0}$ is a scalar entropy value, and S is a finite, acyclic provenance graph recording the derivation history of σ.

Definition 8.2 (Well-Typedness)**.** σ is *well-typed*, written $\sigma \vdash$ valid, if $\mathrm{dom}(M) \subseteq T$: every populated modality is one of the declared required modalities. Note that σ may be well-typed while M is only partially defined on T; well-typedness constrains what may be populated, not what must be.

This single distinction — *declared* modalities T versus *populated* modalities $\mathrm{dom}(M)$ — is what separates a canonical source from a rendering. A source declares that it should eventually support text, audio, and video; a particular saved state of that source may have only text populated. The gap between T and $\mathrm{dom}(M)$ is precisely the space of renderings still owed to the reader.

8.2 Typed Transformations

Definition 8.3 (Rule)**.** A rule is a triple $r : a \rightsquigarrow b$ with entropy budget $\epsilon_r \geq 0$, where a, b are modality tags. Applying r to σ is valid exactly when $a \in \mathrm{dom}(M)$, and produces

$$\sigma \xrightarrow{r} \sigma' = (I,\ T,\ M[b \mapsto r(M(a))],\ E',\ S \cup \{r\}),$$

subject to the entropy guard $E' \leq E + \epsilon_r$.

A rule from PROSE-FULL to VIDEO is a video renderer; a rule from PROSE-FULL to PROSE-SHORT is a summarizer; a rule from EVENT to PROSE-FULL is exactly the "prose is generated from structured events" claim of Section 4, instantiated as a concrete morphism rather than a metaphor.

Theorem 8.4 (Type Preservation)**.** *If* $\sigma \vdash$ valid *and* $\sigma \xrightarrow{r} \sigma'$ *is a valid rule application, then* $\sigma' \vdash$ valid*.*

Proof. By definition of validity, applying $r : a \rightsquigarrow b$ requires $a \in \mathrm{dom}(M) \subseteq T$. The resulting content map is $M' = M[b \mapsto r(M(a))]$, so $\mathrm{dom}(M') = \mathrm{dom}(M) \cup \{b\}$. Since σ is well-typed we have $\mathrm{dom}(M) \subseteq T$; it remains to check $b \in T$. A rule is only declared usable on σ if its target modality is among the required set — this is a side condition on rule declaration, not on application — so $b \in T$ by hypothesis on the rule's admissibility for σ. Hence $\mathrm{dom}(M') \subseteq T$, and $\sigma' \vdash$ valid. □

This is the formal content of "every rendering is a projection of the source, not a separate object": no sequence of rule applications can ever produce a state that has escaped the declared type of the object it was generated from. A video generated from an essay is, by construction, still typed as a rendering of that essay; there is no operation in this calculus that could quietly detach the video from its source and make it free-floating, which is exactly the property YouTube-style publishing lacks.

Corollary 8.5. *For any finite chain* $R = r_1; r_2; \ldots; r_n$ *of valid rule applications, if* $\sigma \vdash$ valid *and* $\sigma \Downarrow_R \sigma_n$*, then* $\sigma_n \vdash$ valid*.*

Proof. Immediate by induction on n using Theorem 8.4 as the inductive step, with the base case $n = 0$ trivial since $\sigma_0 = \sigma$. □

8.3 Rendering as Closure

The closure operator defined below is only as strong as its ability to actually reach every declared modality, and that ability is not automatic. It depends on a renderer existing for each missing modality, which is an assumption about the rule set, not a fact about σ itself. Making this assumption an explicit object, rather than leaving it implicit in the word "declared," is necessary before closure can be stated honestly.

Definition 8.6 (Reach)**.** For a semantic object $\sigma = (I, T, M, E, S)$, define

$$\mathrm{Reach}(\sigma) = \{k \in T : \text{there exists a valid rule chain from some } k' \in \mathrm{dom}(M) \text{ to } k\}.$$

σ is *renderer-complete* if $\mathrm{Reach}(\sigma) = T$.

Renderer-completeness is not guaranteed by well-typedness. A well-typed σ may declare a modality in T for which no rule targeting it has ever been written — a source that promises a video rendering but for which no text-to-video rule exists in the rule set is well-typed and not renderer-complete. This is precisely the situation Section 4 flagged for prose: a declared modality with no known path to it is not a defect in σ, it is an absence in the available rule set, and the two should not be confused.

Definition 8.7 (Closure)**.** The closure operator Q is defined only on renderer-complete objects. For such σ, $Q(\sigma) = \sigma$ if $\mathrm{dom}(M) = T$, and otherwise $Q(\sigma)$ is the result of applying, for every $k \in T \setminus \mathrm{dom}(M)$, some declared rule targeting k from an already-populated modality, until $\mathrm{dom}(M) = T$. Q is undefined on σ for which $\mathrm{Reach}(\sigma) \neq T$.

Proposition 8.8 (Idempotence of Closure)**.** *For renderer-complete* σ*,* $Q(Q(\sigma)) = Q(\sigma)$*.*

Proof. By definition, $Q(\sigma)$ has $\mathrm{dom}(M) = T$. Applying Q again finds no $k \in T \setminus \mathrm{dom}(M)$, since this set is empty, and so returns its input unchanged: $Q(Q(\sigma)) = Q(\sigma)$. □

Restricting closure to renderer-complete objects changes what the idempotence proposition is a proof of. It is not a proof that every declared rendering can be recovered; it is a proof that, conditional on renderers already existing for every declared modality, recovery is well-behaved

and terminates at a stable point. That conditional is doing real work. Whether renderer-completeness holds for a given T — whether, for instance, a faithful video renderer for dense theoretical prose actually exists as a declared rule — is exactly the open problem Section 9.1 shows is solved only for a bounded fragment of prose, and is otherwise unresolved. Closure is therefore best read as a statement about what happens once renderability is available, not as a proof that renderability is available.

Closure is the formal statement of Section 2's "derived layer": a closed object has every declared rendering populated, and nothing about closing an already-closed object does anything further. This is what licenses treating video, audio, and prose-at-depth as *disposable*, for renderer-complete objects. If a rendering is lost or discarded, it is simply an unclosed modality slot, recoverable by re-running Q; nothing about the identity I, the required set T, or the provenance S depended on that rendering having been kept.

8.4 Entropy and the Bound on Drift

Definition 8.9 (Semantic Entropy)**.** For σ with populated modalities $\{M(k)\}_{k\in\mathrm{dom}(M)}$, define

$$E(\sigma) = \sum_{k\in\mathrm{dom}(M)} H(M(k)) \; - \sum_{k\neq k'} \mathrm{MI}\big(M(k), M(k')\big),$$

where H is the entropy of a single modality's content and MI is the mutual information between pairs of modalities.

The subtracted mutual-information term penalizes redundancy: two renderings that say exactly the same thing in different media contribute little beyond their pairwise overlap, while two renderings that are individually coherent but jointly uninformative about one another are, in this accounting, evidence that at least one of them has drifted from the source it claims to represent.

Theorem 8.10 (Entropy Boundedness Under Rule Chains)**.** *For a chain $R = r_1; \ldots; r_n$ with $\sigma \Downarrow_R \sigma_n$,*

$$E(\sigma_n) \;\leq\; E(\sigma) + \sum_{i=1}^{n} \epsilon_{r_i}.$$

Proof. By induction on n. For $n = 0$, $\sigma_n = \sigma$ and the claim is trivial. For the inductive step, assume $E(\sigma_{n-1}) \leq E(\sigma) + \sum_{i=1}^{n-1} \epsilon_{r_i}$. By the entropy guard in the definition of rule application, $E(\sigma_n) \leq E(\sigma_{n-1}) + \epsilon_{r_n}$. Combining the two inequalities gives the claim. □

This is the precise sense in which a source-first system cannot silently accumulate error the way repeated re-uploads or re-transcriptions of a video can: every transformation carries a declared, bounded entropy cost, and the total drift after any finite chain of renderings is bounded by the sum of those declared costs, not by however much distortion happened to occur in an uncontrolled process.

It can help, purely as illustration and not as an additional formal claim, to picture $E(\sigma)$ as tracking the state of a field rather than a static number: a quantity that a well-behaved rule application is required to hold level or push down, and that an undisciplined one is free to let rise unchecked. Nothing in Definition 8.9 or Theorem 8.10 depends on this picture, and the essay makes no claim that semantic drift is governed by anything resembling physical law; the entropy guard is a bound the calculus imposes by definition on ϵ_r, not a conservation principle the world imposes on meaning. The image is offered only because it is a natural way to hold the shape of the boundedness result in mind before returning, in Section 9.2, to the harder question of what E actually measures once symbolic content is left behind.

8.5 Merge and Branching

Definition 8.11 (Guarded Merge)**.** For two objects σ_a, σ_b sharing a common ancestor in their provenance graphs, define the merge $\sigma_a \oplus \sigma_b = \sigma_m$ to be defined only when

$$E(\sigma_m) \leq \max\big(E(\sigma_a), E(\sigma_b)\big) + \epsilon_{\oplus}$$

for some fixed merge budget $\epsilon_{\oplus}$, with $M_m(k)$ reconciled from $M_a(k)$ and $M_b(k)$ by a declared reconciliation rule wherever both are populated, and inherited directly wherever only one is.

Merge is partial by design: two branches that disagree too sharply — whose reconciled entropy would exceed the guard — simply do not merge automatically, and are left as coexisting branches. This is the formal counterpart of Section 6's claim that "neither is a revision of the other": the calculus does not force a choice between competing continuations, it only defines the conditions under which such a choice becomes unnecessary because the continuations were compatible all along.

Proposition 8.12 (Merge Respects the Entropy Bound)**.** *If $\sigma_a \oplus \sigma_b$ is defined, then $E(\sigma_a \oplus \sigma_b) \leq \max(E(\sigma_a), E(\sigma_b)) + \epsilon_{\oplus}$.*

Proof. Immediate: this inequality is the definedness condition of the merge operator itself, not a derived fact. A merge that would violate it is, by definition, not a value the operator produces. □

8.6 Coherence Across Local Views

The remaining requirement is the one this essay has been calling admissibility from Section 1 onward: that renderings generated locally — for one reader, one depth, one branch — must not silently contradict renderings generated for another, when both descend from the same source. This is a gluing condition, and it is naturally stated in the language of sheaves.

Definition 8.13 (Local View Assignment)**.** Let $\mathcal{U}$ be a covering of some index set of contexts (readers, depths, branches) by subsets $\{U_i\}$. A local view assignment is a function G sending each U_i to the rendering of σ appropriate to that context, together with restriction maps $\rho_{U_i U_j} : G(U_i) \to G(U_j)$ whenever $U_j \subseteq U_i$.

Definition 8.14 (Coherence Condition)**.** G is *coherent* if for every pair of overlapping contexts U_i, U_j,

$$\rho_{U_i, U_i \cap U_j}\big(G(U_i)\big) = \rho_{U_j, U_i \cap U_j}\big(G(U_j)\big),$$

and if, whenever a family of local views agrees pairwise on all overlaps, there exists a unique global view $G(\bigcup U_i)$ restricting to each of them.

Theorem 8.15 (Coherence Under Determinism)**.** *If (i) every rendering rule is a deterministic function of its source modality, and (ii) two contexts U_i, U_j derive their local views from the same closed object $Q(\sigma)$, then any local view assignment G satisfies the coherence condition.*

Proof. By (ii), $G(U_i)$ and $G(U_j)$ are both computed by applying declared rules to the same source $Q(\sigma)$. By (i), applying the same rule to the same input yields the same output; hence for any modality k populated in both contexts, $\rho_{U_i, U_i \cap U_j}(G(U_i))(k) = \rho_{U_j, U_i \cap U_j}(G(U_j))(k)$, since both sides are the deterministic image of $M(k)$ under the same rule. Agreement on overlaps established, uniqueness of the glued global view follows because a global assignment consistent with all local restrictions is exactly the restriction of $Q(\sigma)$ itself, and any two global assignments agreeing with every local one must agree with $Q(\sigma)$ pointwise, hence with each other. □

Corollary 8.16. *If coherence fails — if two renderings of the same declared source disagree on their overlap — then by contraposition of Theorem 8.15, either the rules that produced them were not deterministic functions of a shared source, or the two renderings were not, in fact, derived from the same*

closed object. Either diagnosis is actionable: the first identifies a defective renderer, the second identifies an unauthorized fork masquerading as a rendering of the canonical source.

This is the formal cash value of "renderings should not be mistaken for source." Coherence failure is not merely an aesthetic inconvenience to be smoothed over; under this calculus it is a proof-theoretic signal that something has gone wrong upstream, and the corollary above tells you exactly what kind of thing to look for.

8.7 The Limits of Any Rendering

The preceding subsections establish what a source-first architecture guarantees. This one establishes a limit no architecture of this kind, however carefully specified, can remove — and distinguishes that limit sharply from the pathology this essay opened by diagnosing.

The distinction developed here is not a fourth entry alongside the three axes of Section 3. Lineage, depth, and rendering are coordinates: they say which version of a source, compressed how much, in which modality, a request is asking for. Erasure and illusion, by contrast, are not coordinates but failure modes that can occur at any fixed choice of those coordinates, describing what has happened to the provenance behind a rendering rather than which rendering was requested. The two taxonomies meet at exactly one point, and it is worth naming in advance: Proposition 8.23 below shows that moving along the depth axis is the specific mechanism by which an instance of illusion, though never eliminated, can be resolved on demand. Depth is where the coordinate system and the failure mode intersect; lineage and rendering play no comparable role in disambiguation.

Definition 8.17 (Provenance Space)**.** Let $\mathcal{P}$ denote the set of all provenance graphs reachable by finite chains of rule applications from some initial object, under Definition 8.3. Since a legal chain $R = r_1; \ldots; r_n$ may have any finite length n, $\mathcal{P}$ is countably infinite.

Definition 8.18 (Rendering Map)**.** For a fixed modality k, let O_k denote the space of possible outputs of that modality — the set of all texts, audio tracks, or videos of bounded practical length that could occupy $M(k)$. A *rendering map* for modality k is a function $\rho_k : \mathcal{P} \to O_k$ sending a provenance history to the content it produces in that modality.

Any modality with outputs of practically bounded length or duration has $|O_k| < |\mathcal{P}|$: there are only finitely many texts under any fixed length bound, or finitely many videos under any fixed duration and resolution bound, while $\mathcal{P}$ admits chains of unbounded length. This condition holds for every modality this essay has discussed — prose at any depth, audio, video, comics — since none of them is permitted to grow without bound merely to keep pace with an arbitrarily long provenance history.

Theorem 8.19 (No Faithful Rendering)**.** *For any modality k with $|O_k| < |\mathcal{P}|$, every rendering map ρ_k is non-injective: there exist distinct provenance histories $S_1 \neq S_2 \in \mathcal{P}$ with $\rho_k(S_1) = \rho_k(S_2)$.*

Proof. Immediate from the pigeonhole principle: a function from a countably infinite set to a strictly smaller set cannot be injective. □

This theorem is not a limitation of the calculus developed in this essay; it is a limitation on any architecture whatsoever that renders bounded outputs from unboundedly long histories, source-first or otherwise. It is worth being precise about what it does and does not say, because it is easy to mistake it for a restatement of the conflation problem this essay opened with.

Definition 8.20 (Erasure)**.** A rendering pipeline exhibits *erasure* if, after producing $M(k)$ for some modality k, the provenance S used to produce it is discarded and becomes unavailable to future rule applications.

Definition 8.21 (Illusion)**.** A rendering pipeline exhibits *illusion* with respect to modality k if a reader who observes only $M(k)$ cannot, from $M(k)$ alone, determine which of the (necessarily many, by Theorem 8.19) provenance histories in $\rho_k^{-1}(M(k))$ actually produced it.

Proposition 8.22 (What Source-First Architecture Eliminates)**.** *A source-first architecture in the sense of Section 2 eliminates erasure: S is retained regardless of which renderings have been produced from it, per Definition 8.1. It does not eliminate illusion: by Theorem 8.19, any single rendering $M(k)$ remains the image of more than one possible provenance history, and observing $M(k)$ alone never distinguishes among them.*

Proof. Retention of S is immediate from Definition 8.1, in which S is a component of σ that persists across rule applications rather than being consumed by them; nothing in Definition 8.3 removes S upon producing a new populated modality. Non-elimination of illusion follows directly from Theorem 8.19: since ρ_k is non-injective, observing $\rho_k(S)$ alone cannot recover S uniquely, and this holds independently of whether S happens to be retained elsewhere in the system. □

This is the essay's most honest statement of its own scope. Section 1 diagnosed a real and correctable pathology — treating a rendering as though it were the source, so that the source itself becomes unavailable for correction, branching, or citation. That pathology is erasure, and Proposition 8.22 shows the source-first architecture developed here genuinely eliminates it. But no architecture eliminates illusion, because illusion is a cardinality fact about the relationship between unboundedly long histories and boundedly sized outputs, not an engineering shortfall correctable by better design. A reader handed a short summary can never, from the summary alone, rule out every other provenance history that would have produced the identical summary. What a source-first architecture changes is not whether this is true — it is always true — but what a reader can do about it.

Proposition 8.23 (Disambiguation by Depth)**.** *Let $M(k_{\text{short}})$ and $M(k'_{\text{short}})$ be two short-depth renderings, in the sense of Section 3, produced from distinct provenance histories $S_1 \neq S_2$ with $\rho_{k_{\text{short}}}(S_1) = \rho_{k_{\text{short}}}(S_2)$ — an instance of illusion at that depth. If both S_1 and S_2 remain retained, then there exists a full-depth modality k_{full} such that $\rho_{k_{\text{full}}}(S_1) \neq \rho_{k_{\text{full}}}(S_2)$, and this rendering is recoverable by applying Q to the retained source.*

Proof. Take k_{full} to be a modality whose rendering map is the identity on $\mathcal{P}$ restricted to a faithful serialization of S, which exists for any retained S since nothing in Definition 8.1 bounds the length of a full-depth textual rendering the way this essay's Theorem 8.19 bounds practically-sized modalities such as short summaries or fixed-duration video. Since $S_1 \neq S_2$, their full-depth renderings under this map differ. Recoverability follows from Definition 8.7: since S_1 and S_2 remain in dom, closure can be reapplied to either to populate k_{full} on demand. □

This is the precise sense in which retaining provenance is not merely a courtesy to future editors but the mechanism by which illusion, though never eliminated outright, is made *reversible in principle* rather than permanent. A system that erases S after rendering has no such recourse: once S_1 and S_2 have both collapsed to the same short summary and the sources are gone, no amount of staring at the summary recovers which one actually occurred. A system that retains S can always, on request, regenerate a finer rendering and learn something Theorem 8.19 guarantees the coarser one could never have shown. What this essay's architecture offers is not a way around the limit — there is none — but a guarantee that the limit is never made worse than it has to be.

This last point sharpens into a general fact about policies, not just about the two histories S_1, S_2 used above to illustrate it.

Definition 8.24 (Retentive and Erasing Policies)**.** A rendering policy for σ is *retentive* if it preserves S indefinitely across every rule application, as in Definition 8.1. A policy is *k-erasing* if, after populating some modality $k \in T$, it discards S while keeping M, so that no rule chain terminating before the discard remains reconstructible.

Proposition 8.25 (Retention Dominates Erasure)**.** *Fix σ and consider any finite sequence of future disambiguation requests, each of the form "recover a full-depth rendering distinguishing the histories that produced a given illusory output." A retentive policy answers every such request, by Proposition 8.23. A k-erasing policy answers none of them for outputs populated at or before the discard, since the histories needed to distinguish them are no longer available to any rule chain. Retention therefore weakly dominates erasure on this class of requests: it never answers fewer of them, and it answers strictly more whenever at least one such request arrives after the point an erasing policy would have discarded S. Since nothing in Definition 8.1 or Definition 8.3 attaches a cost to retaining S — the entropy guard bounds the cost of* applying *rules, not of storing provenance, and Section 8.8 shows the whole calculus is satisfiable at zero cost throughout — this dominance is not offset by any corresponding advantage erasure holds elsewhere in the formalism.*

Proof. Immediate from Proposition 8.23, which requires only that S_1, S_2 remain in dom at the time of the request; a retentive policy satisfies this by definition for every request, however late it arrives. A k-erasing policy fails this requirement for any request arriving after the discard, since the provenance graph that Definition 8.7's closure operator would need to reapply Q no longer exists in σ. The two policies therefore agree exactly on requests arriving before any discard and diverge on all requests arriving after, giving weak dominance with strict inequality whenever the latter set is nonempty. That retention carries no offsetting cost follows from Definition 8.1, in which S is a component of σ read by the closure and disambiguation machinery but never consumed or bounded by the entropy guard of Definition 8.3. □

What this establishes is narrower than a claim that history is valuable for its own sake, and correspondingly more useful: within this calculus, retention is never a worse policy than erasure on any measure the calculus itself defines, and is strictly better on exactly the measure — future disambiguation — that Theorem 8.19 shows can never be guaranteed in advance. The architecture does not need to argue that provenance matters as a philosophical primitive; it only needs Proposition 8.25, which shows that once illusion is conceded to be unavoidable, retaining the means to resolve any particular instance of it on demand is a dominant strategy rather than a mere preference.

The dominance is stated, and holds, over exactly the costs Definition 8.3's entropy guard prices: the cost of applying a rule. It says nothing about storage, indexing, search, or the operational overhead of keeping a growing provenance graph available in a real deployment, because the calculus does not model those costs any more than it models the currency of an incentive layer. Section 8.8 draws this same boundary for a different reason — separating admissibility, closure, and coherence from any scheme that allocates resources or rewards contribution on top of them — and Proposition 8.25 inherits it rather than needing a separate argument for it. A deployment that finds storage or search costs prohibitive is not a counterexample to the proposition; it is a case where a resource-allocation layer external to the calculus, of exactly the kind Section 8.8 already flags as a separate axiom system, has to decide how much of the dominant policy it can afford to run.

8.8 The Incentive Layer Is a Separate Design Choice

Nothing in the preceding seven subsections requires attaching a cost, stake, or reward to any operation. Well-typedness, closure, entropy bounding, merge, coherence, and the rendering limits above are all satisfiable by a trivial system in which every rule has budget $\epsilon_r = 0$ and

no resource is ever scarce; the proofs above go through unchanged in that degenerate case. It follows that any scheme for allocating attention or rewarding contribution — a market in stakeable credits, a reputation score, a bonding-and-slashing mechanism — is logically independent of the substrate developed here. Such a scheme may be layered on top for reasons of spam resistance or contributor incentive, but it is a separate axiom system, not a consequence of admissibility, closure, or coherence. A reader evaluating this architecture should keep the two questions apart: does the substrate guarantee that renderings stay faithful to their source, and separately, does any incentive layer built on top of it allocate resources in a way that is itself fair, contestable, and free of the attention-optimizing pathologies this essay set out to escape in the first place. The proofs above answer only the first question.

9 Three Problems Made Concrete

Section 8 establishes what the substrate guarantees given certain things are already in hand. Three of those things were asserted rather than shown: that prose admits an event grammar, that the entropy measure E is computable, and that the governance layer can determine canonicality without simply relocating the problem this essay set out to solve. Each is addressed directly below, and in each case the honest answer is more limited than the earlier sections implied.

9.1 A Worked Example of Prose Events

Take a short passage of argumentative prose: *"A teacher tries to increase a student's capacity to evaluate multiple possibilities. A preacher tries to secure commitment to a single viewpoint. Real people usually fall somewhere between the two extremes."* This is exactly the kind of definitional, distinction-drawing prose Section 4 worried a spatial event grammar could not capture. A bounded fragment of it can be captured as follows, using a small closed vocabulary of operations — `introduce_term`, `assert_relation`, `contrast`, `qualify` — each an event in the sense of Definition 8.3, with dependencies recorded explicitly rather than left to be inferred from sentence order:

```
event { id: e1, op: introduce_term, term: "teacher" }
event { id: e2, op: introduce_term, term: "preacher" }
event { id: e3, op: assert_relation, subject: e1,
        relation: "increases", object: "capacity to evaluate
        multiple possibilities", depends_on: [e1] }
event { id: e4, op: assert_relation, subject: e2,
        relation: "secures", object: "commitment to a single
        viewpoint", depends_on: [e2] }
event { id: e5, op: contrast, terms: [e3, e4], depends_on: [e3, e4] }
```

Prose is a rule from this event sequence to PROSE-FULL, exactly as Definition 8.3 describes: the passage above is recoverable by walking e_1 through e_5 in dependency order and rendering each operation through a fixed template. The interesting case is revision. Suppose the passage is later qualified: *"Most educators do some preaching, and most preachers do some teaching."* This is a new event, not an edit to the old ones:

```
event { id: e6, op: qualify, target: e5,
        content: "most educators do some preaching, and most
        preachers do some teaching", depends_on: [e5] }
```

Events e_1 through e_5 are untouched. A diff against this source is the single added node e_6 and its edge into e_5, not a line-level text diff against the rendered paragraph — which is exactly the

property Section 4 wanted and Section 8's provenance graph S was built to support. A renderer that has already cached the prose for e_1–e_5 does not need to re-render them; only the portion of the output downstream of e_6 needs recomputation.

This demonstrates that a bounded fragment of argumentative prose — definitions, asserted relations, contrasts, qualifications — has a workable event grammar. It does not demonstrate that arbitrary theoretical prose does. Analogy, hedged claims, rhetorical questions, narrative exposition, and long inferential chains with implicit rather than declared dependencies are not covered by the four operations above, and it is not obvious that any small closed vocabulary of operations could cover them without either growing without bound or losing the very property — machine-precision — that motivated the event grammar in the first place. What is shown here is a genuine existence proof for a restricted case, not a general solution.

9.2 The Status of the Entropy Measure

Theorem 8.10 is unconditionally valid as mathematics: nothing in its proof depends on how E is computed, only on E being some well-defined real-valued function satisfying the stated guard. The open question is not whether the theorem is true but whether E, as defined in Definition 8.9, can actually be evaluated on real content.

For the worked example above, it can. The events e_1 through e_6 are symbolic objects drawn from a small, fixed operation vocabulary; H over a rule chain can be taken as the Shannon entropy of the empirical distribution of operation types in S, and MI between two symbolic fields — say, the set of terms introduced versus the set of terms qualified — is a standard, tractable computation over finite discrete distributions [1]. Nothing exotic is required.

For genuinely cross-modal content — a video rendering measured against the prose it was generated from — no such procedure is known. There is no principled, agreed definition of the Shannon entropy of a video clip or the mutual information between a video and a paragraph; any number produced for such a pair is, at present, necessarily a proxy computed by some learned multimodal model's embedding similarity, not an instantiation of the formal quantity in Definition 8.9. This matters beyond mere imprecision. A proxy metric is optimizable, and any renderer whose acceptance depends on scoring low E against a learned proxy has an incentive to produce output that games the proxy rather than remains faithful to the source — the standard reward-hacking failure mode for any learned evaluation function substituted for a formal one [2]. Section 8's entropy-boundedness theorem does not prevent this, because the theorem's guarantee is only as strong as the guard $E' \leq E + \epsilon_r$ actually being checked against a faithful E, and a gameable proxy is by construction not faithful in the cases that matter most. The honest position is that Section 8 gives an entropy accounting scheme that is rigorous for symbolic and single-modality content today, and that extending it to cross-modal content is a measurement problem, not a mathematics problem, requiring either a genuine theory of cross-modal information content or an explicit, independently audited proxy whose failure modes are tracked rather than assumed away.

9.3 Canonicality, Defaults, and the Limits of the Substrate

The formal substrate of Section 8 has nothing to say about which branch a new reader is shown by default, and this is not an oversight to be patched but a boundary that should be stated precisely. Two different properties have been running together under the word "canonical," and they need to be separated.

The first is a substrate guarantee: a branch, once created, is never deleted. It remains addressable by its identity I, its provenance S is preserved, and any rule chain that once applied to it can still be applied and cited. This is provided directly by the definitions in Section 8; nothing

in the merge or closure operators removes a losing branch, because "losing" is not a concept the substrate defines. This guarantee is real and is not nothing: it is precisely what YouTube-style demonetization, shadow-banning, or platform-level deletion can violate, and a source-first system of the kind proposed here cannot silently make a branch harder to find in the way a recommendation algorithm can.

The second is a governance choice: which branch, among several that remain equally addressable, is shown to a reader who has not specified a coordinate — the branch returned when someone asks for "the current version" without further qualification. This is naturally written as a pointer-selection function $P : \mathcal{B} \to \mathcal{B}$ over the branch set $\mathcal{B}$, with $P(\mathcal{B}) = b_{\text{default}}$ picking out whichever branch is currently shown by default; "canonical," in this narrower sense, is shorthand for $P(\mathcal{B})$ at a given time, not a property intrinsic to the branch itself. Writing it this way is worth doing even though P is not derived from the formal substrate, because it converts "this branch is canonical" into "this pointer currently resolves to this branch," and the second phrasing makes it obvious that the fact could be otherwise tomorrow without anything about the branch having changed. P itself is not, and cannot be, derived from the formal substrate, because the substrate treats all valid branches symmetrically by design; the choice of P is external to it. Whatever mechanism instantiates P — vote, stake-weighted preference, editorial designation, reputation score — is exactly as capturable by concentrated resources as any comparable mechanism in any other social system, and no result in Section 8 bears on this. A faction with more resources can in principle secure $P(\mathcal{B})$ for its own branch under most plausible instantiations of P, including ones that look procedurally fair.

What the substrate can offer here is narrower than a solution, but not nothing: the pointer-selection process itself can be made a first-class, versioned object subject to the same provenance discipline as everything else. A change to which branch is currently the default is then a citable event with an identity and a timestamp, not a silent update, and the history of default-pointer changes is itself queryable — one can ask not only "what is currently canonical" but "who made it canonical, when, and under what stated selection rule." This converts unaccountable capture into auditable capture. It does not prevent capture, and any claim that it does would be overreach. The central political question the earlier critique identified — how does the system prevent a well-resourced faction from burying competing branches under an ostensibly neutral default mechanism — is not answered by this essay, and it is not answered by the formal substrate developed in Section 8, because it is not the kind of question a type system can answer. It is a governance design problem, it belongs to the PlenumHub layer discussed in the next section, and it deserves treatment on its own terms rather than a deferral disguised as an answer.

9.4 The Citation-Joint Problem: An Unmarked Rule

The preceding three subsections examine gaps between what Section 8 proves and what it can currently deliver. This one examines a gap the formal apparatus has not yet been asked to close at all: nothing in the definitions and propositions of Section 8 prevents a rule application from going undeclared while still changing what a reader takes the source to license.

To state the gap precisely, it helps to name a distinction Proposition 9.3 below already relies on without saying so. The provenance graph S attached to σ is a record of transformations, but a record is not automatically identical to the history it purports to record.

Definition 9.1 (Faithful Provenance)**.** Let S_{recorded} denote the provenance graph actually stored with σ, and let S_{actual} denote the true sequence of transformations that produced σ from its sources. S is *faithful* if $S_{\text{recorded}} = S_{\text{actual}}$, written Faithful$(S)$.

An *unmarked rule*, defined next, is simply the mechanism by which faithfulness fails: a trans-

formation that occurs as part of S_{actual} but never enters S_{recorded}.

Definition 9.2 (Unmarked Rule)**.** Let σ be a semantic object with provenance graph S. An *unmarked rule* is a transformation applied in the course of producing σ' from σ that alters M — by adding content, narrowing a claim, or extending a claim beyond what its cited modality supports — without a corresponding node being added to S_{recorded}.

Theorem 8.4 proves that every *declared* rule application preserves well-typedness. It says nothing about undeclared ones, because the calculus has no way to detect a transformation that was never entered into the provenance record in the first place. An unmarked rule is invisible to the type system not because the type system is weak, but because there is nothing in S_{recorded} for it to fail to preserve.

This is not a hypothetical failure mode. Consider a citation of the form: "Villalobos et al. forecast that public human-generated text may be exhausted for LLM training between 2026 and 2032; looped computation can be viewed as a data-efficiency mechanism responding to this constraint" [3]. The first clause is a legitimate rule application: source paper to claim, faithfully rendered, checkable against the cited text. The second clause is a *different* rule application — an inferential move connecting data-efficiency-in-training to efficiency-of-inference-time-refinement — introduced by the citing author, not licensed by the source, and carrying its own entropy cost in the sense of Definition 8.9: it introduces a claim not present in, and not straightforwardly implied by, $M(\text{source})$. Written as a single sentence with a single citation attached, the two clauses appear to the reader as one rule application inheriting one warrant. They are not. The phrase "can be viewed as" is doing the work of a second, unmarked rule spliced onto the first, and no entry in S_{recorded} distinguishes where the cited claim ends and the citing author's inference begins. It is worth noting that unfaithfulness of this kind can arise in either of two distinguishable ways: from a transformation that occurred but was never logged at all, or from a transformation that was logged as though it were licensed by its input when in fact its output exceeds what that input supports. The essay does not develop a separate entailment calculus to distinguish these two cases formally, since doing so would risk growing a second formal subsystem inside what is otherwise a single, unified account of provenance; both are adequately captured, for the purposes here, as failures of Faithful(S).

Proposition 9.3 (Unmarked Rules Defeat Provenance Without Defeating Type-Correctness)**.** *A chain containing an unmarked rule can satisfy* $\sigma_n \vdash$ valid *under Corollary 8.5 while* $\neg$ Faithful(S).

Proof. Corollary 8.5 establishes well-typedness by induction over *declared* steps in R; an unmarked rule is by definition absent from R and from S_{recorded}, so its presence or absence has no bearing on the induction. σ_n can therefore be well-typed — every populated modality drawn from the declared set T — while $S_{\text{recorded}} \neq S_{\text{actual}}$, which is exactly $\neg$ Faithful(S). Well-typedness is a claim about the shape of M and T; faithfulness of S to the true derivation history is a separate claim, and Corollary 8.5 was never a proof of the latter. □

This matters more than a narrow reading of the type system might suggest, because it identifies exactly where the burden of trust silently shifts. A reader who checks a well-typed σ_n against the calculus and finds no violation may reasonably conclude the source-to-claim chain is intact. Proposition 9.3 shows this conclusion is unwarranted whenever an unmarked rule is present: type-correctness and provenance-faithfulness can diverge, and the calculus as developed in Section 8 certifies only the former. Framed this way, the Citation-Joint Problem is not a one-off anomaly particular to academic citation; it is a special case of provenance drift, the general phenomenon of S_{recorded} silently diverging from S_{actual}, of which an uncredited inferential leap is simply the most legible instance.

The remedy is not a new theorem but a discipline the existing machinery already supports: treat every inferential move attached to a citation, however small, as its own rule with its own

identity in S, rather than letting it travel for free inside the node representing the citation itself. Concretely, the sentence above would be two rule applications, not one: r_1 : source $\rightsquigarrow$ forecast-claim, faithfully bounded by what the source states, and r_2 : forecast-claim $\rightsquigarrow$ design-hypothesis, an inference by the citing author, carrying its own declared entropy budget ϵ_{r_2} and answerable on its own terms rather than borrowing the credibility of r_1. Splitting the two is not pedantry; it is the only way S can do the job Section 8 was built around, which is making clear, for any claim, exactly how far the chain from source to conclusion actually reaches — and where, if anywhere, a reader is being asked to take a step the source itself did not license.

The same explanatory strategy appears outside citation analysis entirely: separate a bundled explanation into its component causes, construct an intervention that removes one component while holding the other fixed, and reassess which component was actually responsible for the observed effect. Diffusion-model training research has used exactly this move to distinguish a data-augmentation effect from a self-supervision effect that had previously been reported as one bundled improvement [4]. The parallel is structural, not substantive — this essay has nothing to say about noise scheduling — but the underlying move, holding one cause fixed and removing the other to see what was actually carrying the effect, is the same discipline the unmarked-rule diagnosis above asks a reader to apply to any citation that bundles a sourced claim with an uncredited inference.

9.5 Confabulated Closure

The unmarked rule of the previous subsection presupposes that some real inference occurred and simply went unrecorded. There is a second, more severe failure mode in which nothing licenses the added content at all — not even an uncredited inference — and it is worth distinguishing from the first, because the remedy for one does not address the other.

Definition 9.4 (Confabulated Closure)**.** A rendering exhibits *confabulated closure* when it presents a modality k as populated, $k \in \mathrm{dom}(M)$, for a σ that is not renderer-complete with respect to k — that is, $k \notin \mathrm{Reach}(\sigma)$ — and the content occupying k was invented to fill the gap rather than derived by any declared rule or left absent.

Definition 8.7 makes Q undefined on objects that are not renderer-complete; the correct behavior when no rule reaches a required modality is for that modality to remain unpopulated, or for the rendering process to say plainly that it could not be produced. Confabulated closure is what happens when a rendering pipeline is asked to populate a modality anyway and, lacking a declared rule, produces plausible-sounding content rather than an empty slot. Unlike an unmarked rule, there is no real transformation to reconstruct here even in principle: a reader who suspects an unmarked rule can in principle work out what inference was likely made and check it against the source, because something was actually inferred. A reader who encounters confabulated closure has nothing to reconstruct, because nothing was inferred; a plausible-sounding value was generated to satisfy the shape of the expected output; with no argument behind it, faulty or otherwise.

This is a real and not merely hypothetical risk for exactly the kind of automated rendering pipeline this essay's Section 6 anticipates. Consider a pipeline asked to compress this essay's own treatment of governance, Section 9.3 above, into a single cell of a comparison table contrasting a conflated publishing era against a source-first one. This essay's own position is that coordination and canonicality are *not* solved by the substrate — that default-branch selection is a governance choice external to the calculus, capturable by concentrated resources, and that the most the substrate can offer is auditable capture rather than prevented capture. A table cell has no room for that qualification; it wants a single confident word. A pipeline under that pressure may produce something like "solved" or "symmetrical," neither of which this essay asserts anywhere, and neither of which follows from Section 9.3 by any argument, sound or

unsound. That is confabulated closure: the modality "one-word governance summary" was never in Reach(σ) for a source that spends a full subsection insisting the question is unresolved, and the correct output was either no cell at all or a cell reading "contested; see full discussion," not an invented resolution.

The practical consequence is that a source-first system needs a declared failure mode for renderers, not merely a declared success mode. Q being undefined on a non-renderer-complete σ is a mathematical fact from Definition 8.7; it becomes a safeguard only if every rendering pipeline built on top of the calculus is required to surface that undefined-ness to the reader rather than silently discharging it with invented content. A closure operator that can fail loudly is a safety property. A rendering pipeline that never admits it has failed, and instead fills every requested slot with something confident-sounding, defeats that property regardless of how carefully Q itself was specified.

9.6 Self-Application: Are the Coherence Conditions Derived or Declared?

Theorem 8.15 rests on two hypotheses — rule determinism, and shared derivation from the same closed object — and both are taken as premises, not derived from anything prior. Nothing in Section 8 grounds determinism of the rule set in something more basic, and nothing derives "same closed object" from anything beneath Definition 8.1's own stipulation of what a semantic object is. A theorem is entitled to its hypotheses; the point is narrower than a defect in the proof. It is that the coherence condition itself — the thing licensing PlenumHub's claim that locally reasonable interpretive acts will not silently contradict one another at scale — is exactly as declared as any other axiom here, not derived from some deeper admissibility geometry that justifies it.

This matters because Section 10 describes PlenumHub as offering governance "emerging from admissibility conditions rather than imposed by centralized moderation," language that implies the conditions are discovered rather than authored. Held to the standard this essay applies elsewhere, that phrasing overclaims. Two earlier cases in this essay's own development were criticized for exactly this move: a governance aggregator whose defining axioms did not uniquely force the aggregator actually used, closed only by an undefended further assumption; and a multi-axis critique gate with no argument that its axes were complete, independent, or necessary rather than merely curated. Theorem 8.15's hypotheses are in the same position. "Rendering rules are deterministic" and "contexts derive from the same closed object" are conditions the architecture requires, not conditions it explains the necessity of. A system built on nondeterministic rules with a probabilistic coherence guarantee, or on some weaker notion of "sufficiently related" closed objects, is not ruled out by anything proven here — it is simply a different design, and nothing above shows the conditions actually chosen are the right ones rather than merely workable ones.

None of this weakens Theorem 8.15 as mathematics; it is correct given its hypotheses. What it weakens is any suggestion that PlenumHub has solved the question of which continuations are admissible, rather than built a calculus that behaves well once a particular notion of admissibility has already been assumed. The honest claim is the smaller one: these conditions are self-consistent, not that they are non-arbitrary. Closing that gap would require showing the conditions are forced by something prior to their own statement, not merely that they happen to work — the same demonstration Section 9.1 asked for and did not supply for prose event grammars, and one that does not currently exist here or elsewhere this essay draws on.

A comparable staging choice, made for unrelated engineering reasons, is visible in recent work compiling natural-language task specifications into lightweight neural artifacts. Program-as-Weights produces a parameter-efficient adapter from a specification in two declared stages

rather than one: a first pass emits an intermediate pseudo-program, and a second pass reads the specification together with that pseudo-program to produce the adapter itself [5]. Nothing about the compilation target forces this split; a single end-to-end pass from specification to adapter is not excluded by the problem, any more than Theorem 8.15's hypotheses are excluded by anything prior to their own statement. The staged design was chosen because a named intermediate type is easier to inspect, debug, and constrain than an opaque one-shot mapping — which is close to Definition 8.3's own reason for typing a rule by its source and target before asking what it computes. The parallel is worth naming for the same reason as Section 9.4's: it is evidence that declaring an intermediate structure, rather than deriving it from necessity, is an ordinary and often correct engineering discipline elsewhere too, not a peculiarity of this essay's own unresolved axioms.

9.7 What This Calculus Guarantees, Enables, and Leaves Open

The preceding six subsections have shown, one at a time, that a formal result proved in Section 8 depends on a condition Section 8 does not itself supply. The pattern is worth collecting once, plainly, rather than leaving it distributed.

Three things are guaranteed outright, for every σ and every rule set, with no further condition to satisfy. Theorem 8.4 shows well-typedness survives any valid rule chain. Proposition 8.8 shows closure is idempotent wherever it is defined. Proposition 8.22 shows erasure is eliminated by construction, since Definition 8.1 never permits a rule application to discard S.

Four things are guaranteed conditionally: true given a stated hypothesis, not true regardless of it. Theorem 8.10's drift bound holds given that E is computable, which Section 9.2 shows only for symbolic content. Theorem 8.15's gluing result holds given rule determinism and shared closed-object derivation, which this subsection has just shown are stipulated rather than derived. Proposition 8.23's illusion-resolution holds given that the histories in question remain retained and a full-depth rendering rule exists for them, neither of which the calculus forces into existence. And Proposition 8.25's retention dominance holds given the calculus's own cost model, which prices rule application but, as stated directly after the proposition, prices neither storage nor search.

Three things the calculus does not address, and has not claimed to. Which branch a reader is shown by default is external to it, per Section 9.3. Whether an inference attached to a citation was actually recorded is undetectable by it, per Proposition 9.3. Whether a rendering pipeline invents content for an unreachable modality rather than reporting failure is unconstrained by it, per Definition 9.4. Each of these three is a property of a rule set, a renderer, or a governance process built on top of the calculus, not a property the calculus's own definitions fix.

A deployment receives the first three guarantees automatically. It earns the middle four only by supplying the conditions they name — a computable entropy measure, deterministic renderers, retained provenance, an affordable storage budget. It must solve the last three itself, by engineering discipline or by a governance layer external to Section 8 entirely, since no theorem in this essay reaches them.

10 Situating the Proposal Within the Stack

Source-first publishing is not a new project sitting alongside PlenumHub and Polyxan; it is what those two projects already imply once applied to video, audio, and prose specifically. The three-layer stack this essay's proposal belongs to can be summarized as follows.

Spherepop supplies event histories, collapse operations, provenance, and execution: the substrate-level fact that identity is an irreversible history of events rather than a persisting essence, and

that nothing is overwritten, only extended. **Polyxan** supplies semantic structure and hypermedia: typed semantic objects rather than isolated documents, bidirectional linking, proof topology, and the claim — inherited explicitly from Project Xanadu [6] — that meaning itself, not merely text, should be addressable, versioned, and traversable in multiple directions. Media quines and sheaf-theoretic worldviews belong here: the requirement that locally reasonable interpretive acts glue together into a globally coherent semantic structure rather than silently contradicting one another at scale [7]. **PlenumHub** supplies coordination, governance, and economic and social infrastructure: identity as provenance and contribution rather than popularity, governance built on top of declared coherence conditions rather than imposed by centralized moderation — though, as Section 9.6 makes explicit, those conditions are themselves stipulated rather than derived, a limitation this essay has not resolved and has no grounds to exempt itself from having flagged in other systems — and, in the later, more explicitly thermodynamic framing, platform pathologies reframed not as bad incentives to be patched but as concentration of semantic capacity and curvature wells that attract attention, calling for redesign of the substrate itself rather than moderation layered on top of it.

All three have at various points been described as programs written in Spherepop calculus, and the recurring slogan across versions of this stack — *meaning as structured objects, identity as provenance, coherence as currency, time as persistence* — is a direct statement of what a canonical-source publishing layer needs to be true in order to work at all. A video generated on demand from a versioned source is a Polyxan object: typed, addressable, linked back to its lineage. Its branching revision history is a Spherepop event log: nothing overwritten, everything collapsed forward from a prior state. And the practice of citing a specific prior state, resolving disputes about which branch is canonical, or crediting a suggested continuation is a PlenumHub governance function: coordination and attribution handled as a property of the substrate rather than imposed by a platform's moderation team after the fact.

Seen this way, the complaint that opened this essay — that YouTube discards its source and treats an uneditable recording as canonical — was never really a complaint about video specifically. Conventional social media generally optimizes for feed-centric competition among posts for attention; the proposal here inherits, from the wider stack, an explicit rejection of that architecture in favor of merge-aware, provenance-aware, event-sourced structures where correctness and coordination are properties of the substrate rather than outcomes fought for within it. The publishing problem turned out to be one more surface on which the same underlying commitment applies: that containers — histories, semantic structures, coordination mechanisms — are prior to and independent of their contents, and that a system built on that premise keeps a wide space of admissible futures open by construction, rather than collapsing it into whatever single frozen artifact happened to get rendered first.

Appendices

A A Minimal Event Grammar for Argumentative Prose

The worked example of Section 9.1 used a restricted event vocabulary sufficient for definitions, assertions, contrasts, and qualifications. This appendix gives that vocabulary in a compact BNF-style form. It formalizes the four operations already used in Section 9.1 and nothing beyond them; it remains, as Section 9.1 states, an existence proof for a bounded fragment of argumentative prose, not a general-purpose grammar.

A.1 Core Syntax

```
<event> ::= introduce_term
          | assert_relation
          | contrast
          | qualify

<introduce_term> ::=
    event {
        id: <event_id>,
        op: introduce_term,
        term: <term>
    }

<assert_relation> ::=
    event {
        id: <event_id>,
        op: assert_relation,
        subject: <reference>,
        relation: <relation>,
        object: <content>,
        depends_on: [<reference>*]
    }

<contrast> ::=
    event {
        id: <event_id>,
        op: contrast,
        terms: [<reference>+],
        depends_on: [<reference>*]
    }

<qualify> ::=
    event {
        id: <event_id>,
        op: qualify,
        target: <reference>,
        content: <content>,
        depends_on: [<reference>*]
    }
```

A.2 Auxiliary Nonterminals

```
<event_id> ::= identifier

<reference> ::= <event_id>

<term> ::= string

<relation> ::= string

<content> ::= string
```

A.3 Dependency Constraints

1. Every event identifier is unique.
2. Every element of `depends_on` must reference an already-existing event.
3. `assert_relation` introduces a directed semantic edge from `subject` to `object`.
4. `contrast` requires at least two referenced events.
5. `qualify` does not replace its target event; it creates a new event whose interpretation is layered onto the target.
6. Event histories are append-only. Existing events are never modified in place.

A.4 Dependency Graph Semantics

Let

$$G = (V, E)$$

be the event graph induced by a source object.

- Each event corresponds to a vertex $v \in V$.
- Each dependency $a \in \texttt{depends\_on}(b)$ induces a directed edge $a \to b$.
- G is required to be acyclic.
- Rendering order is any topological ordering of G.

A.5 Example

```
event { id: e1,
        op: introduce_term,
        term: "teacher" }

event { id: e2,
        op: introduce_term,
        term: "preacher" }

event { id: e3,
        op: assert_relation,
        subject: e1,
        relation: "increases",
        object: "capacity to evaluate multiple possibilities",
```

```
        depends_on: [e1] }

event { id: e4,
        op: assert_relation,
        subject: e2,
        relation: "secures",
        object: "commitment to a single viewpoint",
        depends_on: [e2] }

event { id: e5,
        op: contrast,
        terms: [e3,e4],
        depends_on: [e3,e4] }

event { id: e6,
        op: qualify,
        target: e5,
        content: "most educators do some preaching, and most
                  preachers do some teaching",
        depends_on: [e5] }
```

A prose renderer traverses the graph in dependency order and maps each event type to a corresponding linguistic template. The resulting text is therefore a rendering of the event graph rather than the canonical source itself.

The grammar above specifies what an event is and how events depend on one another; it does not specify how a given event type is turned into English. That step is itself a rule in the sense of Definition 8.3 — a typed transformation from the event sequence to PROSE-FULL — and Definition 8.3 characterizes every rule in this calculus by its source and target type and its entropy budget, never by its internal computation. Two renderers may satisfy the same rule with different template libraries and produce different English for the same event graph; both are equally valid renderings under Definition 8.3, and choosing between them is a renderer design question this appendix leaves open by the same convention the calculus applies to every other rule, not by an oversight specific to prose.

B Notation and Definition Index

This index collects the symbols and named concepts introduced in Sections 8 and 9, in order of first appearance, for reference while reading the later sections that depend on them. It adds no content beyond what is already stated at each cited location.

Symbol / Term	Meaning	Location
$\sigma = (I, T, M, E, S)$	A semantic object: identity, declared modality tags, partial content map, entropy value, provenance graph.	Def. 8.1
I	Immutable identity of a semantic object.	Def. 8.1
T	Finite set of required (declared) modality tags.	Def. 8.1
M	Partial content map from modalities to payloads.	Def. 8.1
E	Scalar entropy value of σ.	Def. 8.1, 8.9

Symbol / Term	Meaning	Location
S	Provenance graph recording σ's derivation history.	Def. 8.1
$\sigma \vdash$ valid	Well-typedness: $\mathrm{dom}(M) \subseteq T$.	Def. 8.2
$r : a \rightsquigarrow b$	A rule: a typed transformation from modality a to modality b.	Def. 8.3
ϵ_r	Entropy budget of rule r.	Def. 8.3
Reach(σ)	Modalities reachable from σ by some valid rule chain.	Def. 8.6
renderer-complete	σ such that Reach(σ) $= T$.	Def. 8.6
Q	Closure operator; populates every declared modality of a renderer-complete σ.	Def. 8.7
H, MI	Per-modality entropy and pairwise mutual information, used to define $E(\sigma)$.	Def. 8.9
$\sigma_a \oplus \sigma_b$	Guarded merge of two objects sharing a common ancestor.	Def. 8.11
$\epsilon_\oplus$	Merge budget bounding the entropy of a guarded merge.	Def. 8.11
G, U_i, $\rho_{U_iU_j}$	Local view assignment over a covering of contexts, with restriction maps.	Def. 8.13
coherent	G satisfies pairwise agreement on overlaps and unique gluing.	Def. 8.14
$\mathcal{P}$	Provenance space: all provenance graphs reachable by finite rule chains.	Def. 8.17
ρ_k	Rendering map from provenance histories to outputs of modality k.	Def. 8.18
O_k	Space of practically bounded outputs for modality k.	Def. 8.18
erasure	Discarding S after producing $M(k)$, making it unavailable to later rules.	Def. 8.20
illusion	Inability to recover which provenance history produced an observed $M(k)$.	Def. 8.21
retentive / k-erasing	Policies that preserve S indefinitely, or discard it after populating k.	Def. 8.24
S_{recorded}, S_{actual}	The stored provenance graph versus the true derivation history.	Def. 9.1
Faithful(S)	$S_{\text{recorded}} = S_{\text{actual}}$.	Def. 9.1
unmarked rule	A transformation altering M that is absent from S_{recorded}.	Def. 9.2
confabulated closure	Content occupying a modality outside Reach(σ), invented rather than derived or left absent.	Def. 9.4
$P : \mathcal{B} \to \mathcal{B}$	Pointer-selection function choosing the default branch shown to a reader.	§9.3

References

[1] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[2] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[3] P. Villalobos, A. Ho, J. Sevilla, T. Besiroglu, L. Heim, and M. Hobbhahn. Will we run out of data? Limits of LLM scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*, 2022 (revised 2024).

[4] D. Jiang, M. Wang, H. Yang, and J. Wang. From SRA to Self-Flow: data augmentation or self-supervision? *arXiv preprint arXiv:2607.02508*, 2026.

[5] W. Zhang, L. Hotsko, W. Kim, P. Nie, S. Shieber, and Y. Deng. Program-as-Weights: a programming paradigm for fuzzy functions. *arXiv preprint arXiv:2607.02512*, 2026.

[6] T. H. Nelson. Xanalogical structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use. *ACM Computing Surveys*, 31(4es), 1999.

[7] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer-Verlag, 1992.